IN THE CLAIMS

1. (Currently amended) In a computer system, an improved method for developing a Web application, the method comprising:

providing a Web application development framework, said framework including an abstract command tag that predefines at least some generic Web application activities;

specifying at least one custom action that is desired to be performed by a Web application ~~under development~~;

creating an object-oriented programming language (OOPL) class that extends the abstract command tag for providing execution logic for said at least one custom action, in addition to pre-existing logic that supports said at least some generic Web application activities, thereby creating a corresponding customized command tag that is capable of being embedded within a Web page~~, wherein said customized command tag includes the ability to conditionally execute said specified at least one custom action based on run-time conditions~~;

upon execution of the Web application including an embedded customized command tag in a Web page, invoking the customized command tag for conditionally executing said specified at least one custom action based on run-time conditions.

2. (Original) The method of claim 1, wherein said run-time conditions include run-time parameters specified during invocation of the customized command tag.

3. (Original) The method of claim 2, wherein said run-time parameters are specified via Hypertext Transport Protocol (HTTP) parameters, during invocation of the customized command tag.

4. (Original) The method of claim 1, wherein said abstract command tag comprises an abstract base class.

5. (Original) The method of claim 1, wherein said abstract command tag includes an abstract execute method.

6. (Original) The method of claim 5, wherein said abstract execute method is overridden during creation of the customized command tag, for defining a customized execute method providing specific runtime execution logic for the customized command tag.

7. (Previously Presented) The method of claim 5, wherein creation of the OOPL class that extends the base class includes providing an implementation for the abstract execute method.

8. (Original) The method of claim 1, wherein said customized command tag includes an ability to conditionally affect application flow based on results obtained from a specified action.

9. (Original) The method of claim 8, wherein application flow is affected by routing to a particular Web page.

10. (Original) The method of claim 8, wherein said result obtained is either success or failure.

11. (Original) The method of claim 10, wherein application flow is directed to a first page if a success is obtained as the result, and is directed to a second page if a failure is obtained as the result.

12. (Original) The method of claim 8, wherein said application flow includes routing to a different page than is currently displayed in a user's browser.

13. (Original) The method of claim 1, wherein said generic Web application activities include error recording.

14. (Original) The method of claim 1, wherein said generic Web application activities include filtering of requests.

15. (Original) The method of claim 1, wherein said generic Web application activities include page routing.

16. (Original) The method of claim 1, wherein said generic Web application activities include activities that may be predefined before application execution.

17. (Original) The method of claim 1, wherein said customized command tag is invoked when an end user activates a link that points to a Web page containing the customized command tag.

18. (Original) The method of claim 17, wherein said link comprises a Uniform Resource Locator (URL).

19. (Currently amended) The method of claim 17, wherein said Web page containing the customized command tag comprises a ~~JSP (JavaServer Page) compatible~~ Web page generated using dynamic scripting capability.

20. (Currently amended) The method of claim 1, further comprising:

compiling the ~~JSP-compatible~~ Web page generated using dynamic scripting capability into a servlet, said servlet corresponding to said created OOPL class that extends the abstract command tag.

21. (Currently amended) A Web application framework comprising:

specification of an abstract command tag that predefines at least some generic Web application activities;

a programming environment for:

(i) specifying at least one custom action that is desired to be performed by a Web application under development, by supporting creation of an object-oriented programming language (OOPL) ~~OOPL~~ class that extends the abstract command tag for providing execution logic for said at least one custom action, thereby creating a corresponding customized command tag that is capable of being embedded within a Web page, wherein said customized command tag includes the ability to conditionally execute said specified at least one custom action based on run-time conditions; and

(ii) enabling embedding of the customized command tag in a Web page of the Web application;

wherein execution of the Web application includes invocation of the customized command tag for conditionally executing said specified at least one custom action based on run-time conditions.

22. (Original) The framework of claim 21, wherein said run-time conditions include run-time parameters specified during invocation of the customized command tag.

23. (Original) The framework of claim 22, wherein said run-time parameters are specified via Hypertext Transport Protocol (HTTP) parameters, during invocation of the customized command tag.

24. (Original) The framework of claim 21, wherein said abstract command tag comprises an abstract base class.

25. (Original) The framework of claim 21, wherein said abstract command tag includes an abstract execute method.

26. (Original) The framework of claim 25, wherein said abstract execute method is overridden during creation of the customized command tag, for defining a customized execute method providing specific runtime execution logic for the customized command tag.

27. (Previously Presented) The framework of claim 25, wherein creation of the OOPL class that extends the base class includes providing an implementation for the abstract execute method.

28. (Original) The framework of claim 21, wherein said customized command tag includes an ability to conditionally affect application flow based on results obtained from a specified action.

29. (Original) The framework of claim 28, wherein application flow is affected by routing to a particular Web page.

30. (Original) The framework of claim 28, wherein said result obtained is either success or failure.

31. (Original) The framework of claim 30, wherein application flow is directed to a first page if a success is obtained as the result, and is directed to a second page if a failure is obtained as the result.

32. (Original) The framework of claim 28, wherein said application flow includes routing to a different page than is currently displayed in a user's browser.

33. (Original) The framework of claim 21, wherein said generic Web application activities include error recording.

34. (Original) The framework of claim 21, wherein said generic Web application activities include filtering of requests.

35. (Original) The framework of claim 21, wherein said generic Web application activities include page routing.

36. (Original) The framework of claim 21, wherein said generic Web application activities include activities that may be predefined before application execution.

37. (Original) The framework of claim 21, wherein said customized command tag is invoked when an end user activates a link that points to a Web page containing the customized command tag.

38. (Original) The framework of claim 37, wherein said link comprises a Uniform Resource Locator (URL).

39. (Currently amended) The framework of claim 37, wherein said Web page containing the customized command tag comprises a ~~JSP (JavaServer Page) compatible~~ Web page generated using dynamic scripting capability.

40. (Previously Presented) The framework of claim 21, further comprising:

a tag library descriptor providing an association between a customized command tag and its corresponding OOPL class.

41. (Currently amended) An improved method for Web application development, the method comprising:

providing a Web-based application development framework built from a set of object-oriented programming language (OOPL) ~~OOPL~~ classes, said framework providing:

a non-programmatic tag framework that implements the functionality of the application framework when executing within a ~~JavaServer Pages (JSP)~~ Web page generated using dynamic scripting capability;

tag-based Web application objects controlling program flow, executing user commands, representing application business objects, and constructing output;

a non-programmatic tag framework that accesses data for logical business objects and allows page designers to specify an action to be performed;

enabling the embedding of the tag-based Web application objects in a Web page of a Web application; and

execution of the Web application, including invoking the tag-based Web application objects.

42. (Original) The method of claim 41, wherein said non-programmatic tag framework includes detecting and reporting error conditions to either a page developer or an end user running a Web browser.

43. (Currently amended) The method of claim 41, wherein said set of OOPL classes run in a Java Virtual Machine (JVM™), wherein the JVM™ is an interpreter that interprets OOPL bytecodes into machine code.

44. (Currently amended) The method of claim 43, wherein said JVM™ is running at a Web server site.

45. (Original) The method of claim 41, wherein specified actions can be filtered by matching a tag attribute with an HTTP request parameter.